

Table of Contents

1. Directory trees	2
1.1 Enable the extension	2
1.2 Write a tree	2
1.3 Configure indentation and icons	3
1.3.1 Fix indentation errors	4
1.3.2 A report project	4
1.4 Reference	4
1.5 Customise the appearance	5
1.5.1 Generated HTML	5

1. Directory trees

`prodockit.tree` turns an indented list of folders and files into a directory tree. Use it to show readers where files belong in a project.

1.1 Enable the extension

Add an empty extension table to `zensical.toml` when the defaults are suitable:

```
[project.markdown_extensions."prodockit.tree"]
```

1.2 Write a tree

Write an indented file and folder list inside a `tree` block. The result tab shows how the same source is presented to a reader.

Markdown

```
/// tree
docs/ - the documentation source tree
  index.md - the home page
  stylesheets/ - style sheets for the website and PDF
    pdk.css - managed website and PDF defaults
    extra.css - project website and PDF changes
    pdk-pdf.css - managed PDF defaults
    print.css - project PDF changes
  zensical.toml - project configuration
///
```

Result

```

📁 docs — the documentation source tree
├── 📄 index.md — the home page
├── 📁 stylesheets — style sheets for the website and PDF
│   ├── 📄 pdk.css — managed website and PDF defaults
│   ├── 📄 extra.css — project website and PDF changes
│   ├── 📄 pdk-pdf.css — managed PDF defaults
│   └── 📄 print.css — project PDF changes
└── 📄 zensical.toml — project configuration
```

The complete block starts and ends with a three-slash fence. Its body follows three rules:

- **Indentation is the structure.** Two spaces per level by default.
- **A trailing / means a directory.** Anything else is a file. Nothing else marks one, so nothing else can disagree with it.
- **- starts a description**, and it is optional. The spaces are required, which is what keeps `harvard-cite-them-right.csl` in one piece.

1.3 Configure indentation and icons

There are no additional `prodockit.tree` settings in `zensical.toml`. Configure each tree inside its Markdown block. Put the options directly below `/// tree`, indent them by at least four spaces, and leave a blank line before the listing:

[Table 25.1](#) shows the tree options for indentation and file or folder icons.

Table 1.1. Configure indentation and icons

Option	Default	What it does
<code>indent</code>	<code>2</code>	How many spaces one level costs. Set <code>4</code> for a listing written that way.
<code>directory_icon</code>	<code>':lucide-folder:'</code>	Icon shortcode placed before every directory.
<code>file_icon</code>	<code>':lucide-file:'</code>	Icon shortcode placed before every file.

This example changes both the indentation and the icons:

Markdown

```
/// tree
  indent: 4
  directory_icon: ':octicons-file-directory-16:'
  file_icon: ':octicons-file-16:'

docs/
  index.md
///
```

Result

```

└─ docs
   └─ index.md

```

1.3.1 Fix indentation errors

A listing is read for its shape, so an entry attached to the wrong parent is a diagram that is wrong and looks right. Rather than guess, the build stops:

```

TreeError: indent of 3 is not a multiple of 2: 'index.md'
TreeError: indented 2 levels at once: 'index.md'

```

1.3.2 A report project

Use the block for a structure readers need to understand. This example shows the main files supplied by `prodockit-template`:

```

└─ docs
   └─ index.md — cover page
   └─ section1.md — first report section
   └─ acronyms.md — acronym definitions
   └─ glossary.md — glossary definitions
   └─ references.md — generated reference list
   └─ stylesheets
      └─ pdk.css — managed website and PDF defaults
      └─ extra.css — project website and PDF changes
      └─ pdk-pdf.css — managed PDF defaults
      └─ print.css — project PDF changes
└─ tools
   └─ mermaid — diagram renderer
   └─ mathjax — maths renderer
└─ zensical.toml — navigation and extension configuration
└─ references.bib — bibliography source

```

The package's own source-code map now lives under [Contributor internals](#).

1.4 Reference

Use [Table 25.2](#) to look up the block syntax, options, and nesting markers.

Table 1.2. Reference

Syntax or option	Purpose
<code>/// tree</code>	Open or close a directory tree
A trailing <code>/</code>	Mark an entry as a directory
<code>- description</code>	Add an optional description

Syntax or option	Purpose
<code>indent: 4</code>	Use four spaces for each level instead of two
<code>directory_icon: '...'</code>	Choose the directory icon
<code>file_icon: '...'</code>	Choose the file icon
<code>attrs: {...}</code>	Add an id, class, or other attribute to the tree

The `tree` block follows the same fence, option, and nesting rules as [PyMdown Blocks](#).

1.5 Customise the appearance

The extension adds stable class names that you can target in your project's CSS style sheet. Prodockit's managed `docs/stylesheets/pdk.css` contains the default styles used by the examples on this page. Put project-specific changes in `docs/stylesheets/extra.css`, which is loaded afterwards.

1.5.1 Generated HTML

The extension provides structure; your project supplies the appearance:

```
<div class="prodockit-tree">
  <ul>
    <li class="tree-directory">
      <span class="tree-icon">...</span>
      <span class="tree-name">docs</span>
      <span class="tree-note">the documentation source tree</span>
    <ul>
      <li class="tree-file">
        <span class="tree-icon">...</span>
        <span class="tree-name">index.md</span>
      </li>
    </ul>
  </li>
</ul>
</div>
```

Stable class names include `.prodockit-tree`, `.tree-directory`, `.tree-file`, `.tree-icon`, `.tree-name`, and `.tree-note`.

`docs/stylesheets/pdk.css` carries the maintained starting point. When overriding it in `docs/stylesheets/extra.css`, keep the rail and stub positioned from one shared measurement, so changing indentation cannot pull them apart, and stop the last child's rail at its own stub rather than continuing past the final entry.